CAMPUS
DE EXCELENCIA
INTERNACIONAL

Graduado en Matemáticas e Informática

Politécnica de Madrid

Escuela Técnica Superior de
Ingenieros Informáticos

TRABAJO FIN DE GRADO

Improving flexible databases searches using Machine learning

Autor: Alejandro Hernández Munuera

Director: Susana Muñoz Hernández

MADRID, ENERO 2016

“Stay hungry, stay foolish.”
- Steve Jobs

AGRADECIMIENTOS

En primer lugar, agradecer a mi familia el amor que he recibido durante toda mi vida y que estoy seguro voy a seguir recibiendo. Nunca podre compensarles todo lo que me han dado, los principios que me han enseñado y las oportunidades que he tenido gracias a ellos. Lo único que me queda es amarles tanto como pueda cada día de mi vida. Especial mención

También dar gracias a mis amigos y amigas, aquellos con los que he pasado mas tiempo del que debería. Ellos son culpables en gran manera de la persona y el ingeniero que soy. Gracias por enseñarme tantas cosas, hacerme disfrutar del día a día y sobre todo por abrirme la mente de mil maneras.

También mencionar al Instituto Veritas y la UPM por la educación que me han dado, ha sido excepcional y seguiré trabajando para aprovechar el conocimiento obtenido y seguir desarrollándome como ingeniero y persona

Por ultimo, agradecer a todos los que han participado en el proyecto de FleSe, un gran proyecto del que me siento orgulloso de haber participado. Espero de mi trabajo sea del agrado de todos ellos y que esta gran idea y proyecto sigan desarrollándose.

Gracias.

ABSTRARCT

The main objective of this project has been to introduce machine learning in the application FleSe. FleSe is a web application that makes fuzzy queries over databases with precise information, using defined criteria to define the fuzzy concepts used by the queries. The application allows the users to change and custom these criteria. On this point is where the machine learning would be introduced, so FleSe learn from every new user customization of the criteria in order to generate a new default value of it.

The secondary objectives of this project were get familiar with web development and web design in order to understand the how the application works, as well as refresh and improve the knowledge about fuzzy logic and logic programing.

During the realization of the project and after the study of the results, I realized that clustering the users in different groups makes the difference between this new version of the application and the previous. This conclusion follows the next idea, we can use an algorithm to introduce machine learning over the criteria that people have, but the problem is the diversity of opinions and judgements that exists, making impossible to generate a unique correct criteria for all the users. In order to solve this problem, before using the machine learning methods, we cluster the users in order to make groups that have the same opinion, and afterwards, use the machine learning methods to precise the default criteria of each users group.

The future improvements that could be important for the next versions of FleSe will be to control better the behaviour of the plserver file, that cost many troubles at the beginning of this project and it also generate important errors in the previous version. The file plserver allows the web application to use Ciao-Prolog, a logic programming language that control and manage all the fuzzy logic. One of the main problems with plserver is that when the user uploads a file with errors, it will block the thread and when this happens multiple times it will start blocking all the requests. Oriented to the customer, would be important as well to allow FleSe to manage and work with SQL databases instead of store the data in the Prolog files.

Another possible improvement would that the cluster algorithm would be based on different criteria depending on the fuzzy concepts that the selected Prolog file have. This will generate more meaningful clusters, and therefore, the default criteria offered to the users will be more precise.

KEY WORDS

Fuzzy logic, Ciao-Prolog, FleSe, machine learning, cluster

RESUMEN

El objetivo principal de este proyecto ha sido introducir aprendizaje automático en la aplicación FleSe. FleSe es una aplicación web que permite realizar consultas borrosas sobre bases de datos nítidos. Para llevar a cabo esta función la aplicación utiliza unos criterios para definir los conceptos borrosos usados para llevar a cabo las consultas. FleSe además permite que el usuario cambie estas personalizaciones. Es aquí donde introduciremos el aprendizaje automático, de tal manera que los criterios por defecto cambien y aprendan en función de las personalizaciones que van realizando los usuarios.

Los objetivos secundarios han sido familiarizarse con el desarrollo y diseño web, al igual que recordar y ampliar el conocimiento sobre lógica borrosa y el lenguaje de programación lógica Ciao-Prolog.

A lo largo de la realización del proyecto y sobre todo después del estudio de los resultados se demuestra que la agrupación de los usuarios marca la diferencia con la última versión de la aplicación. Esto se basa en la siguiente idea, podemos usar un algoritmo de aprendizaje automático sobre las personalizaciones de los criterios de todos los usuarios, pero la gran diversidad de opiniones de los usuarios puede llevar al algoritmo a concluir criterios erróneos o no representativos. Para solucionar este problema agrupamos a los usuarios intentando que cada grupo tengan la misma opinión o mismo criterio sobre el concepto. Y después de haber realizado las agrupaciones usar el algoritmo de aprendizaje automático para precisar el criterio por defecto de cada grupo de usuarios.

Como posibles mejoras para futuras versiones de la aplicación FleSe sería un mejor control y manejo del ejecutable plserver. Este archivo se encarga de permitir a la aplicación web usar el lenguaje de programación lógica Ciao-Prolog para llevar a cabo la lógica borrosa relacionada con las consultas. Uno de los problemas más importantes que ofrece plserver es que bloquea el hilo de ejecución al intentar cargar un archivo con errores y en caso de ocurrir repetidas veces bloquea todas las peticiones siguientes bloqueando la aplicación. Pensando en los usuarios y posibles clientes, sería también importante permitir que FleSe trabajase con bases de datos de SQL en vez de almacenar la base de datos en los archivos de Prolog.

Otra posible mejora basarse en distintas características a la hora de agrupar los usuarios dependiendo de los conceptos borrosos que se van ha utilizar en las consultas. Con esto se conseguiría que para cada concepto borroso, se generasen distintos grupos de usuarios, los cuales tendrían opiniones distintas sobre el concepto en cuestión. Así se generarían criterios por defecto más precisos para cada usuario y cada concepto borroso.

PALABRAS CLAVE

Lógica borrosa, Ciao-Prolog, FleSe, aprendizaje automático, agrupaciones.

INDEX

FIGURES INDEX

1. INTRODUCTION

The main purpose of this project has been to introduce machine learning in a web application that makes fuzzy queries over databases with precise information. The application where the machine learning algorithms have been added is FleSe(Flexible Searches in Databases). This application has been implemented and developed by Victor Pablos Ceruelo and It allows the users to make queries base on fuzzy concepts over databases with precise information.

More precisely, this project focuses on the criteria that define the fuzzy concepts that are used to make the queries. The fuzzy concepts have default criteria but the users can modify and personalize them, so the results of the queries are adapted to their opinions and criteria. This project has introduce the concept of machine learning over these criteria, so FleSe learn from each new personalization of the users to change and improve the definition of the default criteria. The idyllic result of this new functionality would be that FleSe would give a personalize criteria to each user without the need of been changed by them.

1.1. CONTEXT

Nowadays, data science has become one of the most important and innovative fields inside of the computer science world. This is due to the quantity of information that is available and the meaningful conclusions and knowledge that can be obtained from it through a proper analysis.

Following this idea, it is important to develop and research about new ways to consult this data, allowing an easy manage of it. For this cause, the functionality of making queries over databases with precise data using fuzzy concepts is a new approach to consult information, allowing the users consult large databases with precise data using concepts and expressions that are similar to the ones the are used to manage.

Apart from this functionality that FleSe already had, this project has introduce machine learning over the criteria used by the fuzzy queries. The concept of machine learning has a tight relation with a correct analysis of the information, in order to generate meaningful knowledge from the data is being collected. This relay on a correct and deep analysis of the information that is stored and the ability to analyse new data efficiently, relating this new data with the knowledge already obtained. Owing to this process, the correctness and precision of the conclusions and knowledge generated will be improving while the application is getting more data from the users.

In this particular case, FleSe will use the personalization that the users do over the criteria in order to learn from their opinions and offer a more precise and personalize default criteria to each new user.

1.2. FUZZY LOGIC

The fuzzy logic is one of the important concepts that the web application FleSe manages in order to be able to consult the databases using fuzzy concepts.

The computers and machines usually work with information and data precisely defined where there is no room for doubt. All the information they process and the concepts they deal with are perfectly defined, not allowing subjective opinions and therefore, all the machines understand the concepts and the reality in the same way.

Unlike the machines and the computers, the humans usually manage fuzzy concepts, subjective information without a precise definition strictly established. This means that two people could have different definitions of the same concepts and at the same time the can understand each other when they are exchanging information and talking about the concept without misunderstandings. It is also possible that dealing with the same concepts and logic, two people could conclude different results. For instance, the concept of closeness, everyone has a similar definition of this concept, but at the same time there exists clear differences between these opinions. One person can think that 30 meters is close and another person could think that is not close at all. Even the opinion of a person can change under different circumstances or over time.

Although the fuzzy concepts seems to not offer much clearness, the human has developed a fuzzy logic that allows us to manage this concepts and deal with them to obtain and infer new knowledge from those fuzzy concepts, and then make decisions taking into account all that information.

In order to allow the machines and computer to manage and deal with these fuzzy concepts and get them close to the fuzzy logic that the human use, a membership function is defined for each fuzzy concept. This membership function will assign a membership coefficient or truth-value between zero and one to each of the values belonging to the fuzzy concept domain, zero meaning that the value from the domain doesn't belong to the concept. For instance, the membership function of the closeness concept could assign a truth-value of 0.75 to the distance twenty meters, meaning that twenty meters is close with a truth-value of 0.75. Then a relation between the precise data (the domain of the concept) and the fuzzy concept is established thanks to the membership function.

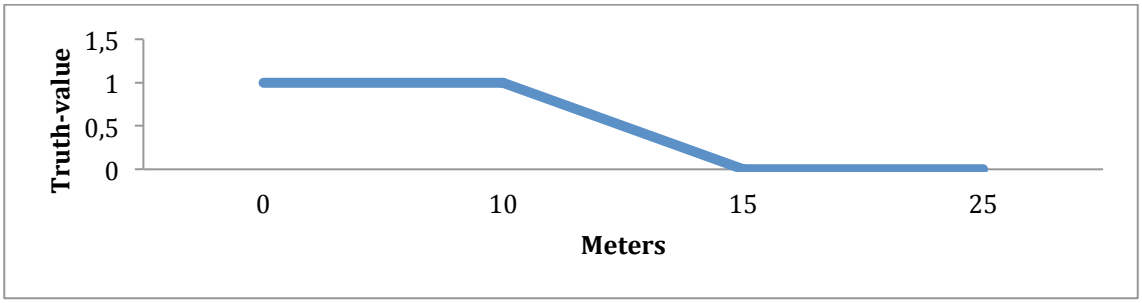


Figure 1: Membership function for the closeness fuzzy concept

In the same way, a list of fuzzy logic rules are declared, with its truth-value, to allow the machines and the computer make fuzzy logic operations with the concepts defined and take decisions with the knowledge obtained from them.

1.3. OBJECTIVES

The main objective of the project has been to introduce machine learning over the criteria that defined the fuzzy concepts used by FleSe to make the queries over the databases. The result would be to achieve the point where each user of the web application wouldn't have the necessity to change the default criteria because FleSe will already offer a personalize criteria to every user.

In order to get this result, we will implement two different algorithms to introduce machine learning. After the implementation will be important to obtained as much customizations of the criteria as possible in order to obtain enough data for the algorithms to generate precise default criteria for each user.

One of the algorithms will generate a default criterion from a list of criteria customizations using machine learning concepts. This result will be coherent and representative of all the customizations of the users.

The second algorithm will cluster the users depending on the characteristics they have. This is trying to make groups of users with similar opinions and definitions of the fuzzy concepts, in order to make the algorithms more efficient and offer a more customize criteria to each user. As the application uses fuzzy concepts to make the queries, this algorithm will not relay in specific and simple comparisons between the users as just cluster the users whose characteristics are equal, and at the same time, it will include machine learning concepts, improving the clusters as it is getting new users included in the groups, redefining the groups with more precision.

The secondary objective of this project has been to get familiar with the web application, as well as with the tools needed to develop it and make it work. The installation of the web application in the personal laptop needed to make research of different tools and computer science fields. The testing part of the application also needed to review and refresh concepts from artificial intelligence, fuzzy logic and logic programming in order to create and change logic programing files to be used by FleSe.

2. PREVIOUS WORK

In this chapter the tools that has been used for this project will be introduced, and the concepts that FleSe deal with will be explained. The first point that will be explained are the logic programing language, the library RFuzzy and the fuzzy logic. The second part of the chapter will explain how FleSe used these tools.

2.1. CIAO PROLOG AND THE LIBRARY RFUZZY

Due to the intention of being working with fuzzy concepts and fuzzy logic, appears the need of using Ciao Prolog. Ciao Prolog allows implementing, compiling and executing programs written in a logic programming language. Prolog, unlike other programing languages, allows defining functions as the logic relation between the input and the output, being the best tool to implement logic programs and develop fuzzy logics. However, in order to implement programs with definitions of fuzzy concepts, fuzzy inference rules and, as a result, a fuzzy logic, it is necessary to add the RFuzzy library to the Ciao Prolog libraries.

The library RFuzzy introduces the concept of membership functions and the truth-value. It allows defining inference rules with its corresponding truth-value, fuzzy quantifiers, fuzzy concepts and its membership functions. With these tools we are able to implement programs with a fuzzy logic and a database, in order to make queries using fuzzy concepts and fuzzy quantifiers over a database with precise information.

2.2. FLESE

FleSe is a web application developed using the programming language Java. The application connects its web users through an accessible and simple user interface to the functionalities that Ciao Prolog and the library RFuzzy provide. FleSe requires the user to login in the system using a social network as Twitter, Facebook or Linked-In or others platforms.

When the user is already registered in the system, he has the possibility to upload his own Prolog files, containing the database, the fuzzy concepts, the quantifiers and the fuzzy inference rules he wants to work with.

If the users don’t want to work with his own files, the the rest of users files will be available. Finally, the main functionality of the application, make the fuzzy queries. The application will ask to choose the specific file that contains the fuzzy logic and the database the user wants to work with. Depending on the concepts and quantifiers defined in the file, the user could make different queries using those concepts.

FleSe also offer to the user to the possibility to personalize the definitions of the concepts contained in the file being used, in case of disagreeing with the default definition or if the results obtained from the queries are not correct according to their criteria. The user would personalize the definition of these concepts changing its membership function.

The membership function of each fuzzy concept is defined assigning the truth-value to a few values from the concept domain, and doing interpolation to obtain the truth-value for the rest values of the domain. So the user have the opportunity to change the truth-value of those few values from the domain, and therefore, make queries using the personalization of the criteria in order to obtain results according to their opinions and definitions of the concept.

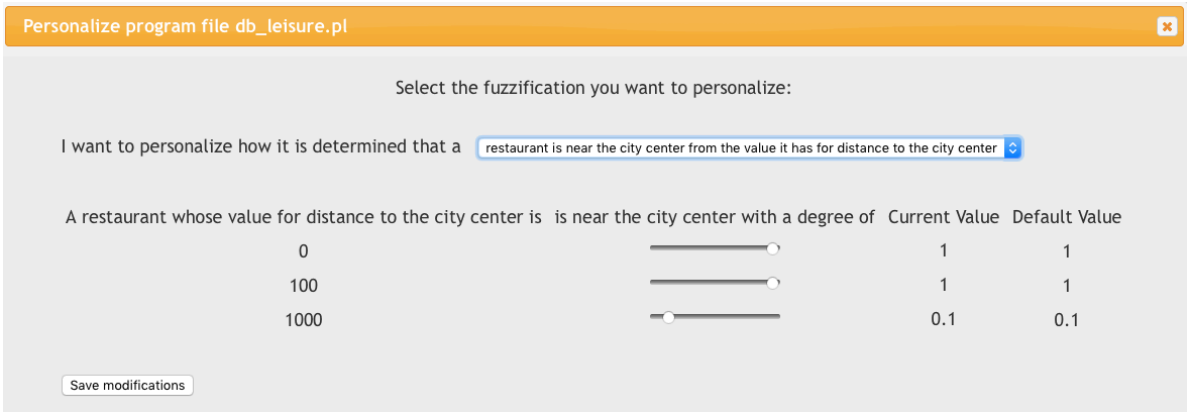


Figure 2: Screenshot of Flese offering to change the criteria of a concept

In the figure 1 can is shown how FleSe shows to the user how the membership function of the fuzzy concept is defined, giving truth-values to the values 0 meters, 100 meters and 1000 meters from the domain of the concept closeness. In order to change the criteria, the user would give different truth-values to these values, and then FleSe will obtain the truth-value for the rest values from the domain interpolating the truth-values received from the user.

3. PROJECT DEVELOPMENT

The work done along this project is divided in three parts, the first part is the installation of the tools and the configuration of FleSe, the second part is the research done about the machine learning algorithms and the final part is the implementation and the inclusion of the algorithms in the web application.

3.1. INSTALATION

Due to different causes that will be explained ahead, the installation process took much more time than it was expected at the beggining.

3.1.1. CIAO PROLOG, EMACS AND THE LIBRARY RFUZZY

The first step of the installation process was to download and install the text editor Emacs 24 (https://www.gnu.org/software/emacs/) and Ciao Prolog (http://ciao-lang.org) to being able to compile, execute and easily develop programs implemented with a logic programming language. These installations were done in the last version of the Macintosh operative system, and there were not big problems due to the detailed installation guides that both programs have, but Ciao Prolog will cause problems afterwards due to the fact that FleSe requires functionalities that Ciao Prolog stable versions don't have, only the version under development.

The following step was to install the Ciao Prolog plug-in in the text editor Emacs 24 in order to make easier and more efficient programming and developing Prolog programs using the text editor. This plug-in allow the user to compile, execute and debug Prolog programs quickly and easily, and introducing changes in the programs is much easier using this tool. On this step, be using the last version of the Macintosh operative system was an important problem. Due to this fact, the installation process was different from the installation guide and was necessary to create and change configuration files of the text editor in order to run this plug-in everytime the text editor was opening a Prolog file.

Once Ciao Prolog was installed and the plug-in was working, the library RFuzzy had to be installed. The library was obtained from the professor Susana Muñoz Hernández and the installation was as simple as copy the library inside the Ciao libraries folder and changes its permissions.

3.1.2. FLESE INSTALLATION

In order to run the application FleSe on the personal computer, the installation of the software Eclipse (http://eclipse.org) was needed and configure it properly downloading the plug-in Webtools for Eclipse. Then download the server Tomcat (http://tomcat.apache.org/download-70.cgi), install it and add it to Eclipse.

When Eclipse and the server are configured and installed, the project FleSe (obtained from the professor Susana Muñoz) has to be imported to Eclipse and the jar servlet.api (http://www.java2s.com/Code/Jar/s/Downloadservletapijar.htm) need to be added. Due to the version of the operative system, the file paths defined in the FleSe configure file might be wrong, so they have to be changed.

After doing the entire installation process and due to the installed version of Ciao Prolog and the version of the operative system, the file plServer that belongs to Ciao Prolog was generating errors. This file has the responsibility to connect the application FleSe with Ciao Prolog. Because of the version of the operative system the plServer file had not been created. Contacting with Ciao user support and using the Ciao compiler "ciaoc", plServer was created with all the files needed to its execution. However, the plServer was still crashing and as a result FleSe was not allowing to make any query over the files. This last problem was because of the Ciao version. The file plServer does not work properly in the stable versions of Ciao, but it does in the version under development, so the solution was to reinstalled Ciao with this last version.

3.1.3. TESTING AND USE OF THE TOOLS

Once the tools are installed is recommendable to get familiar with them in order to understand the how FleSe works and to be able to create Prolog files that contains databases and fuzzy logic to be consulted by FleSe.

At this point was important to work with the text editor Emacs 24 and the Ciao Prolog plug-in, creating simple logic programs using the logic programming language. Due to these smalls programs the user get used to the grammar of the language and the way the functions need to be expressed. After the simple programs, the user should start implementing logic programs using the library RFuzzy and write these programs with the format and style that FleSe ask for, defining databases, fuzzy concepts, quantifiers and fuzzy inference rules. At this point was of great help the material and examples received from Susana Muñoz.

After these steps, the user is able to create his own files containing fuzzy logic and databases with the style required by FleSe. The use of the web application is easy once you have Ciao Prolog files to consult. The user just has to choose a Prolog file that contains the fuzzy concepts and the quantifiers he wants to use to make the queries and change the criteria of the concepts in order to see the difference of the results when applying different customizations of the criteria.

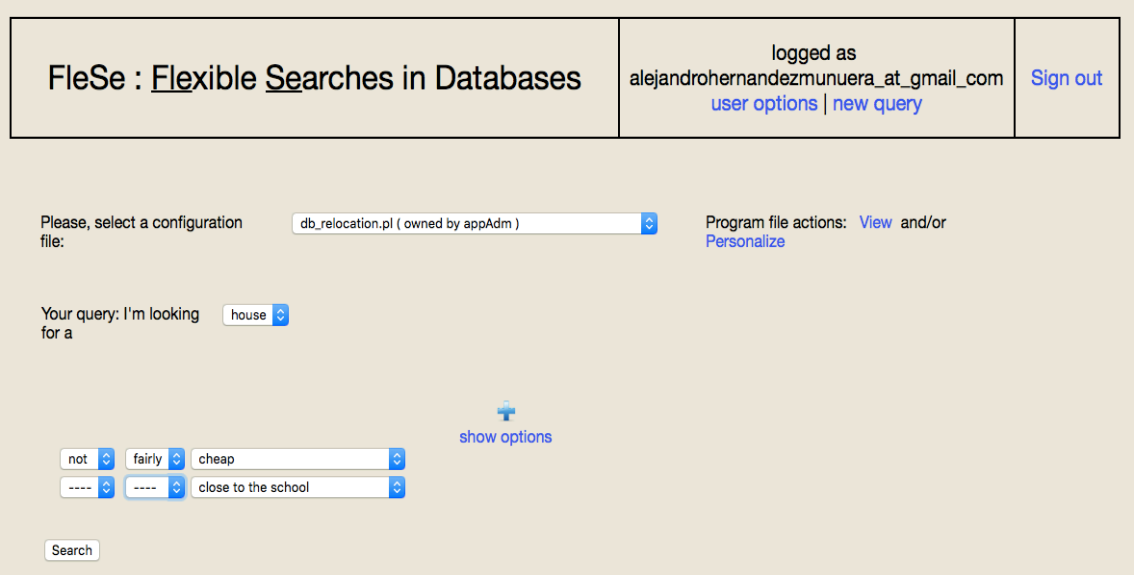
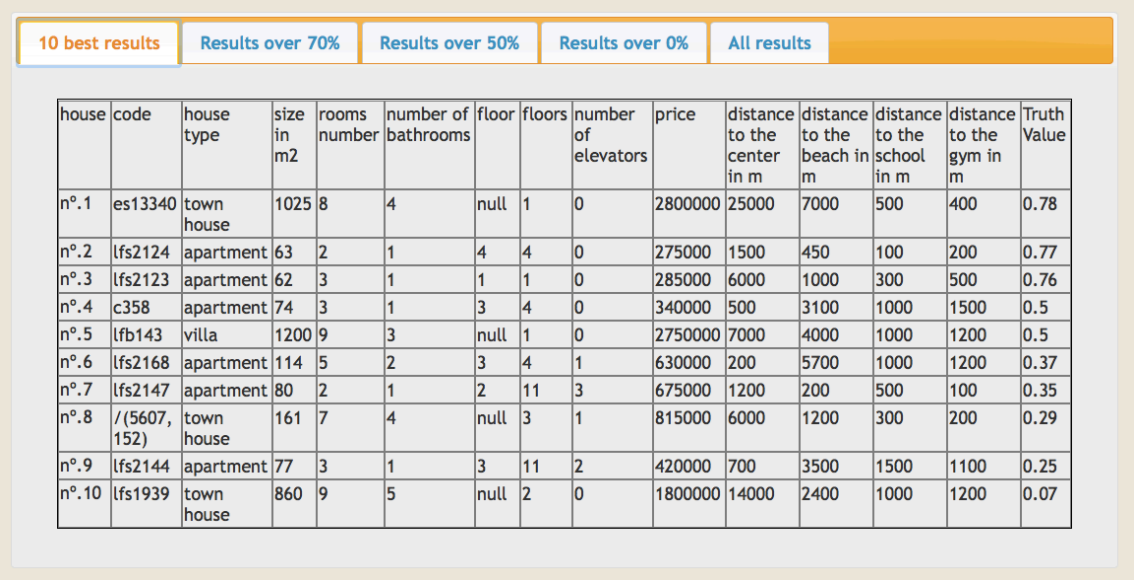


Figure 3.1: Screenshot of FleSe offering to make a query

In the figure 3.1 can be seen how FleSe first offer the option to select the file to work with, and then, use the fuzzy concepts contained in the file to make the query. In this case the concepts used are cheap and closeness. FleSe will show the results to the user as is shown in the figure 3.2.



house	code	house type	size in m2	rooms number	number of bathrooms	floor	floors	number of elevators	price	distance to the center in m	distance to the beach in m	distance to the school in m	distance to the gym in m	Truth Value
nº.1	es13340	town house	1025	8	4	null	1	0	2800000	25000	7000	500	400	0.78
nº.2	lfs2124	apartment	63	2	1	4	4	0	275000	1500	450	100	200	0.77
nº.3	lfs2123	apartment	62	3	1	1	1	0	285000	6000	1000	300	500	0.76
nº.4	c358	apartment	74	3	1	3	4	0	340000	500	3100	1000	1500	0.5
nº.5	lfb143	villa	1200	9	3	null	1	0	2750000	7000	4000	1000	1200	0.5
nº.6	lfs2168	apartment	114	5	2	3	4	1	630000	200	5700	1000	1200	0.37
nº.7	lfs2147	apartment	80	2	1	2	11	3	675000	1200	200	500	100	0.35
nº.8	/(5607, 152)	town house	161	7	4	null	3	1	815000	6000	1200	300	200	0.29
nº.9	lfs2144	apartment	77	3	1	3	11	2	420000	700	3500	1500	1100	0.25
nº.10	lfs1939	town house	860	9	5	null	2	0	1800000	14000	2400	1000	1200	0.07

Figure 3.2: Screenshot of FleSe showing the results of a query

3.2. ALGORITHMS

In order to introduce machine learning over the criteria personalization of the users two algorithms will be used one after another. The first algorithm will cluster the users in groups trying to get groups of users with similar criteria. The second algorithm will produce a representative criteria of all the customizations that the users belonging the group have made.

3.2.1. K-MEANS

In order to cluster the users in different groups we will use the algorithm k-means. This algorithm is used to cluster a set of points in groups, so the first need is to represent each user as a point in a space of n-dimensions.

With this purpose, first we have to choose a list of user characteristics that we will use to classify them. Once we have defined the list of characteristics that is going to be used, every possible value of each characteristic need to have a numeric value in order to make possible to apply the algorithm over them. Then, a function to give a numeric value to each characteristic has to be defined. The numeric value has to be representative for each value of the characteristic, for instance, with the characteristic "country", it wouldn't have sense to assign the numeric value 1 to Spain, 2 to Japan and 16 to France, because these values don't represent the relation that the countries have between them and it can cause wrong results from the k-means algorithm.

After defining these functions, each user will be represented by a set of numeric values with a length of n, being n the number of characteristics selected before. Therefore each user will be represented by a point in a n-dimensions space. Once we have our list of points, we can apply the algorithm k-means.

The purpose of this algorithm is to include every user in a group that represents his criteria, so the first step will be define the groups where the users are going to be included. In order to define the groups, representative users and their characteristics have to be defined. For instance, to define two groups of users, we could define a first representative user with the following characteristics: country: France, age: 20; and the second: country: USA, age:50. This two users will represent a group of young European and old American, completely different characteristics and therefore, completely different opinions. We will obtain the points of these representatives using the functions we have commented before to get numeric values from their characteristics.

Figure 3.3: Initial hypothetic situation for the algorithm K-means

Once we have the representatives of the groups and the users as points in the space of n-dimensions, as shown in the figure 3.3, the Euclidian distance between each user and each representative will be calculated. Then, every user will belong to the group that has the representative point closer to the user point. Once every user belongs to a group, the representative of the groups will be recalculated as the mean of the points that belongs to the group. The process will start again deciding again which is the group of every point depending on the Euclidian distance to the representatives. The process will iterate until the representatives of every group have the same value as the obtained in the previous iteration or until it reach a defined limit of iterations.

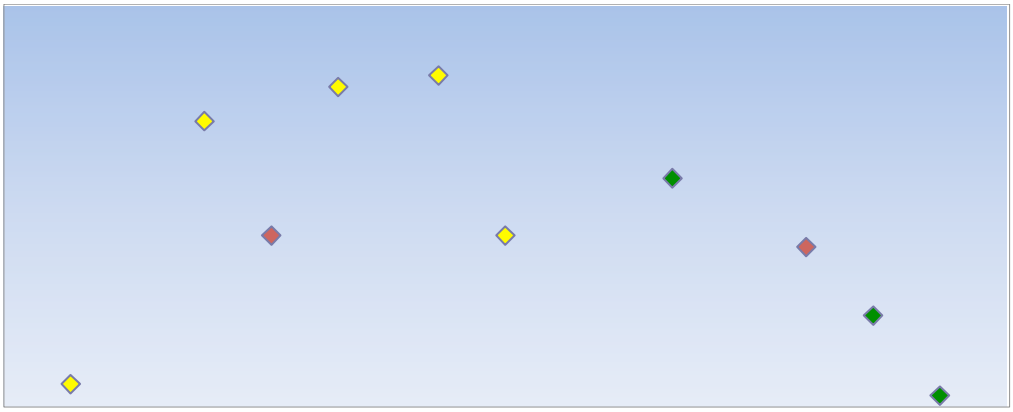


Figure 3.4: Result of the K-means algorithm

3.2.2. MACHINE LEARNING ALGORITHM

This second algorithm will be applied over the criteria customizations of the users who belong to the same group as the user who is using FleSe.

As the customizations of the criteria are an array of numeric values, we will use the personalization as a point in a space of n-dimensions. It is important to realize that every numeric value of the personalization has to be between zero and one, because they are a truth-value. As a consequence, the length of every axis of the space will be always one.

Due to the fact that the space of the personalization will have n-dimensions, being n the number of values that FleSe requires to change the definition of the concept, and that every dimension will always have short length, we could divide this space efficiently in order to cluster all the customizations. Therefore the fist step of this algorithm will be divided the space equally.

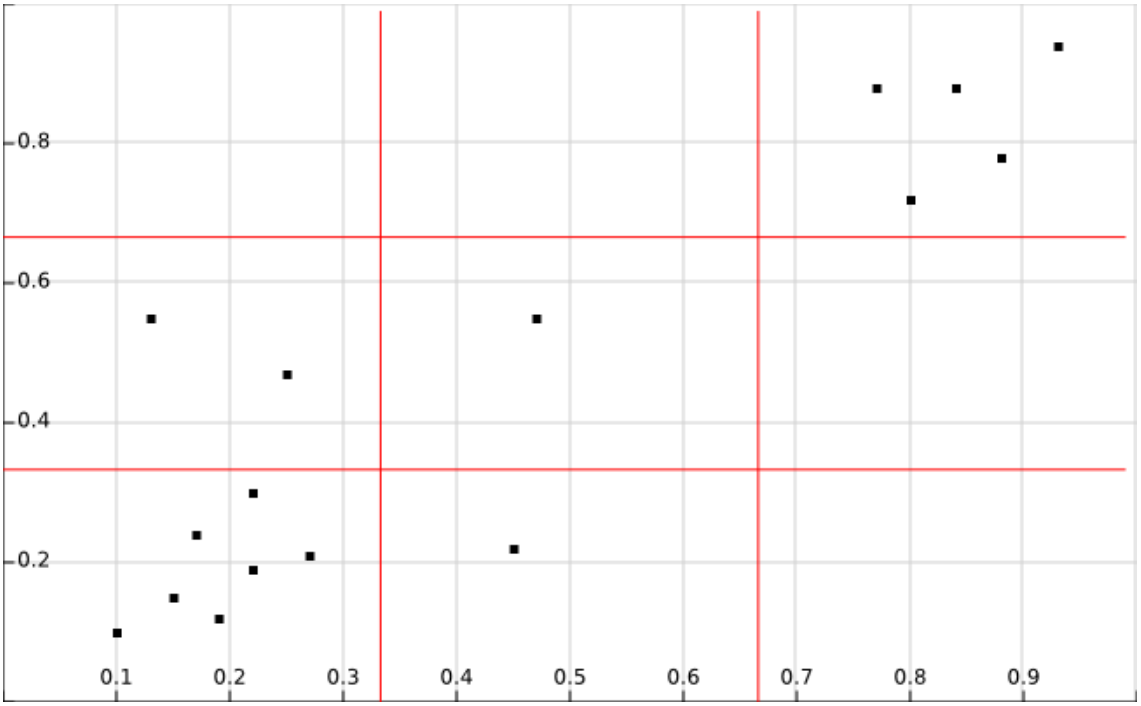


Figure 3.5: Example of a criteria customization space

Once the space is divided, the algorithm will only take into account those customization that belong to the division of the space with more points inside.

Finally the algorithm will return the median of the points selected before.

3.3. IMPLEMENTATION

The implementation process follows three steps. First, make possible to access the characteristics of any user when FleSe needs to execute the k-means algorithm. Second, implement the k-means algorithm in order to include every user in one of the groups defined. Third step was implement the machine learning algorithm to offer every user a personalize default criteria.

These new methods will be executed when a user choose a file, in order to generate the right default criteria values for the user, and when a new personalization is done in order to calculate the new default criteria. So every time a default criteria has to be updated, the method update() from the class FuzzificationsManager will be called.

3.3.1. ACCESS TO USER INFORMATION

The web application already had a class, LocalUserInfo, where the information of the user is stored. So, in this class, I implemented a function called getCharactericticArray in order to obtain the array of numeric values that represent the value of the characteristics of the user. At this moment, the function only take into account the following characteristics: the platform that the user has used to log-in and also the language that is set up his platform account. In case of not being able to access one of those characteristics, the corresponding value in the array will be zero, and will be taken into account when executing the k-means algorithm.

Is also important to know that Flese does not have any file or structure where it stores the information of every user. The application just stores their usernames, the uploaded files and the customizations of the criteria that they have done. The default criteria and the customizations that the users do are stored inside of the Prolog file where the fuzzy concept belongs, as well as the default criteria values. Due to this way of performing, I decided to store the characteristics array of the user with their customizations of the criteria in order to know the characteristics of the owner of every customization when reading from the Prolog file

The last version of FleSe was storing the criteria customizations writing inside the Prolog files the name of the user, the name of the concept customizaes and new truth-values that the user has assign the concept. For now on, the array with the numeric values of the user characteristics will be stored with that information, so will be available anytime the k-means algorithm need to be execute.

3.3.2. K-MEANS IMPLEMENTATION

This method will not be called in case of having none characteristics of the user available. So the method will check if all the numeric values from the actual user characteristics array are zero, in that case the method will return the customization of every user without applying any filter.

In case the at least one of the values of the characteristics array is different from zero, the algorithm K-means will be applied over all the criteria customizations belonging to the corresponding fuzzy concept.

Taking into account the situation where not all the characteristics of the user who is using the application are available, the method will discard those missing characteristics from the characteristic array of every user. The result will be a list of the characteristic arrays from every user without the characteristics that the user does not have.

In the class FuzzificationsManager are also defined the characteristics of the sample users that represents the groups where we will cluster all the personalization. On this version of FleSe there are two representatives, the first will be a user that use Twitter for the login for FleSe and his language is English and the characteristics of the second sample user are Google as login platform and Spanish as language.

With these two representatives and the list of all the characteristics arrays filtered, we can apply the algorithm k-means as explained in the chapter 2. The result is two different lists of characteristics arrays, each one would be the group of users that belong to the group represented at the beginning by the sample users. Finally it will return the list that has the characteristics array of the user who is using FleSe.

In order to generate useful output for the next machine learning algorithm, this method will go through all the criteria customization and select the ones whose characteristics array are in the list returned before.

3.3.3. MACHINE LEARNING ALGORITHM IMPLEMENTATION

This method is implemented in the class FuzzificationAlgorithms and its input is a list of criteria customization generated by the method described above.

As we have explained before a personalization will be an array of numeric values, representing the truth-values assigned to a list of values from the fuzzy concept domain. The length of the array will depend on the number of values from the domain that FleSe allow the user to change. In the example from the figure 3.6, the length of the array obtained from the personalization will be four.

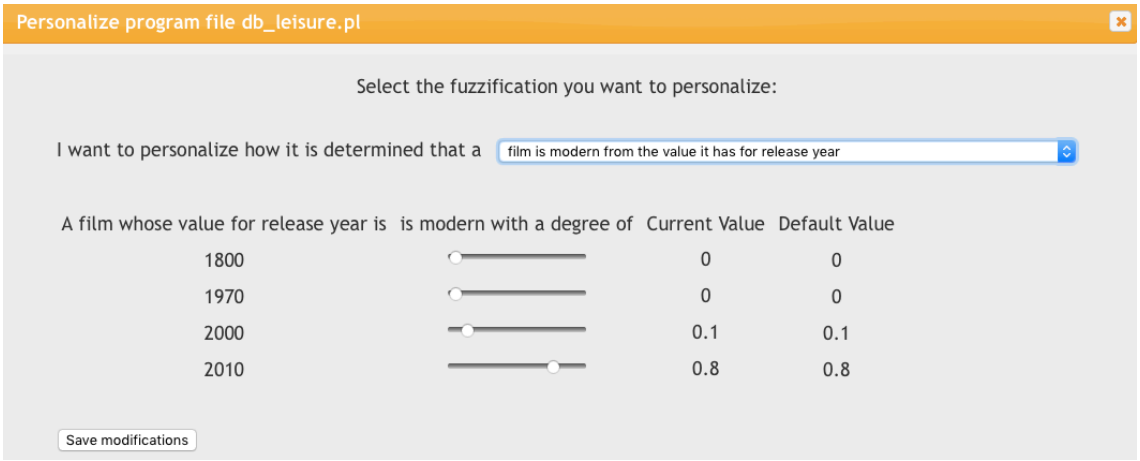


Figure 3.6: Screenshot of Flese offering to change the criteria of a concept

As said before, the criteria customization will be represented by points in a space of n-dimensions, being n the length of the criteria customization array.

The first step of the method is to divide the space where the personalization points belong. Each of the axis of this space will goes from zero to one, because the values from the customization arrays have to be greater than or equal to zero and less than or equal to one. Each axis from the space will be divided at least in two and in ten as the most due to small size that the space has. In order to know in how many parts each axis will be divided, we divide the number of customization we have as input by the number of dimensions that the space has and we round down the result.

When we have the space divided. We select the division which has more customization points inside, in case or having multiple spaces with the same amount we will select all of them.

The output of the method will be the median of all the points that belong to the subspaces selected before. This result will be an array of truth-values that will be used to define the default criteria for the according fuzzy concept.

4. RESULTS AND CONCLUSSIONS

The problem at this point is that in order to generate a good quantity of results it is needed to have many users using the new version of FleSe. Due to different causes, it was not possible to publish the new version of FleSe on Internet and get feedback from real users, but I created fake users in different platforms with different characteristics in order to use FleSe through these accounts. Although we didn't have a good quantity of users, the differences in the results and the improvements can be appreciated even with a small list of users.

4.1. RESULTS FROM PREVIOUS VERSION

The previous version of FleSe was already applying a machine learning algorithm over the criteria customizations of the users. This method was defining as the default criteria the median of all the customizations, that would be the median of all the points that represent the criteria customizations of the users.

The first problem that the method had was that it was not returning exactly the median. The algorithm was just returning the exact median when the number of customizations received as input was odd. The cause of this problem was that when the algorithm was calculating the median of a list of values, it first sort the list and then it will return the value that is in the middle of the array in case the length of the list is odd, but when the length of the list is even the method return the first position value of the two middle positions of the list. For instance, in case we have the list of numbers 2,4,6,0,8 the algorithm would return the value 4, and in case the list was 2,4,6,0,8,10 the result would be 4 as well.

The other problem is that in many situations this method will return a criteria customization that will not represent the knowledge that can be obtained and learnt from all the customizations of the concept. For instance, in the situation represented in the figure 4.1 the algorithm will return the personalization represented by the point (0.5,0.5). This result doesn't represent neither the groups of users whose criteria customization are in the zones with more points. It would have more sense to return a criteria customization that belongs to one of those zones.

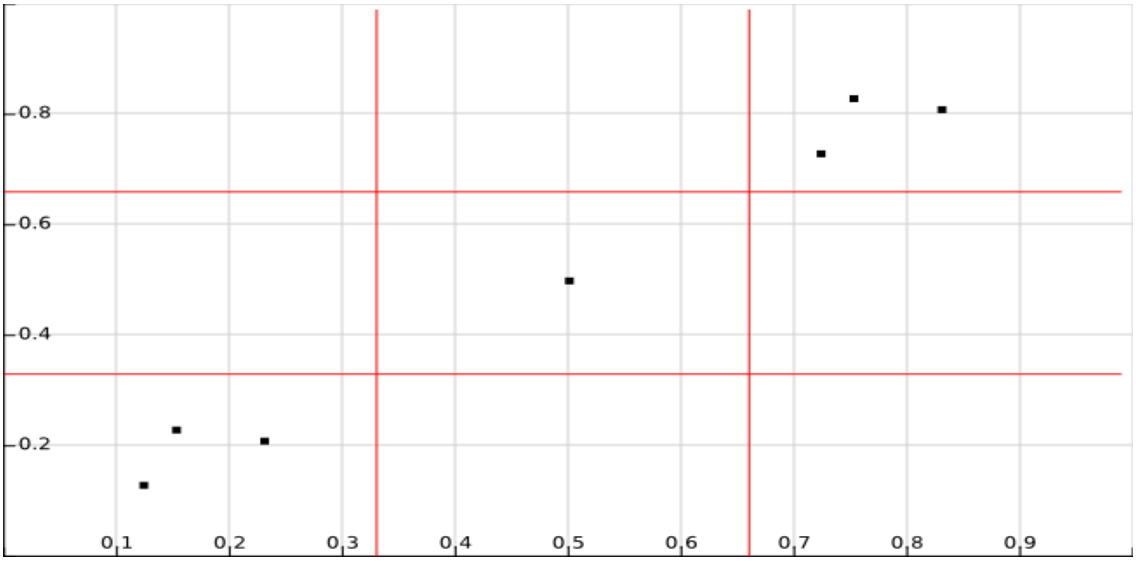


Figure 4.1: Example of a criteria customization space

4.2. RESULTS WITHOUT CLUSTERING

In this chapter we will analyse the results of applying the new machine learning algorithm described before, without using the k-means algorithm to cluster the users.

This method solved the problem the last version had with the median and, due to selection of the zone with more points inside before calculating the median, some problems derived from situations as the one shown in the figure 4.1 are solved as well.

However, applying this algorithm still shows some problems in its result. The first problem is conceptual. The point is that we can't offer the same default criteria and expect that everyone agree with it. There can be several opinions of the same concept, and as a consequence we cannot expect to get a unique default criteria that fits every user. The figure 4.2 shows two groups of users with completely different opinions or criteria about the same concept. In this situation the algorithm will return the median of the criteria customization whose points belongs to the zone with more points, not taking into account that the user who is using the application may be part of the users whose criteria belong to the other zone

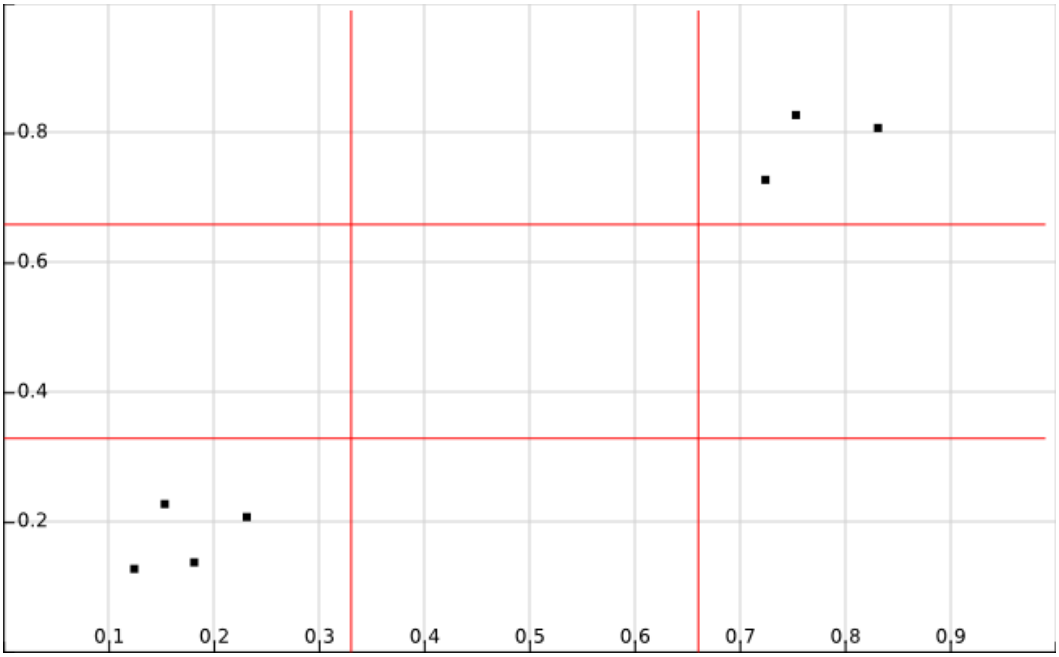


Figure 4.2: Example of a criteria customization space

The second problem will be related with having more than one zone with the maximum number of customizations inside. This situation is represented by the figure 4.3. The dark points are the customization done by the users and the green point is the result of the method. The method will select the two zones with more points inside and then calculate the median of them, which will result in a criteria that doesn't represent any of the user customizations groups.

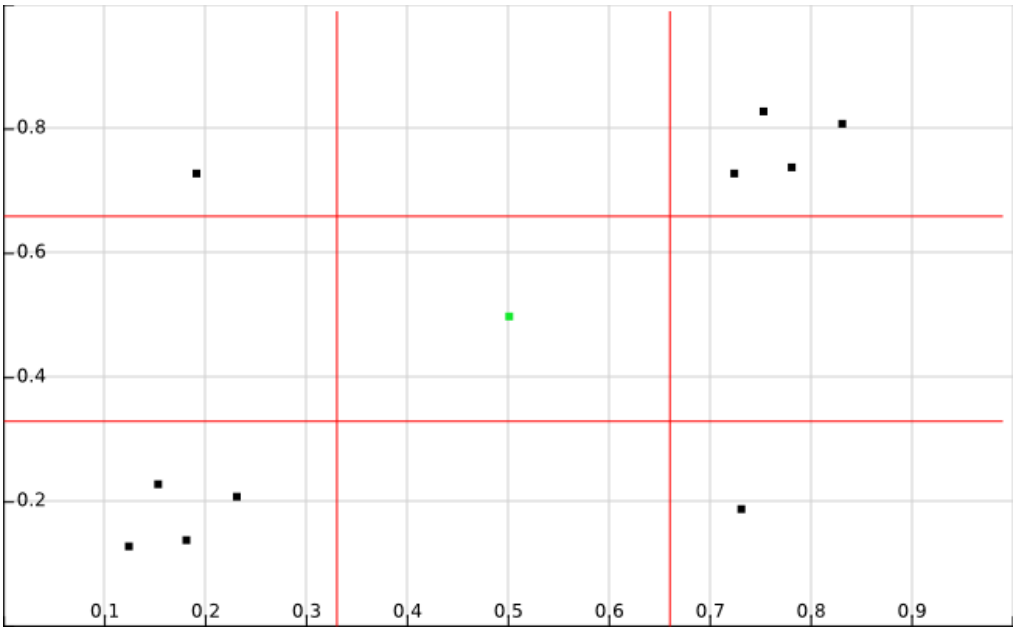


Figure 4.3: Example of a criteria customization space

4.3. RESULTS WITH CLUSTERING

The method k-means will cluster every user in a group and will return the customizations of the group where the user that is using the application belongs. The purpose of this method is that the users of the groups generated will have similar criteria.

So with these clusters the problems commented above will be solved. The situation represented by the figure 4.4 shows how the method has clustered all the users. The green points represent the customizations of the users clustered in the group one and the blue point are the customization of the users clustered in the second. Then FleSe will apply the machine learning algorithm over the customizations that belongs to the group where the user who is using the application has been clustered. Avoiding the problems we had before

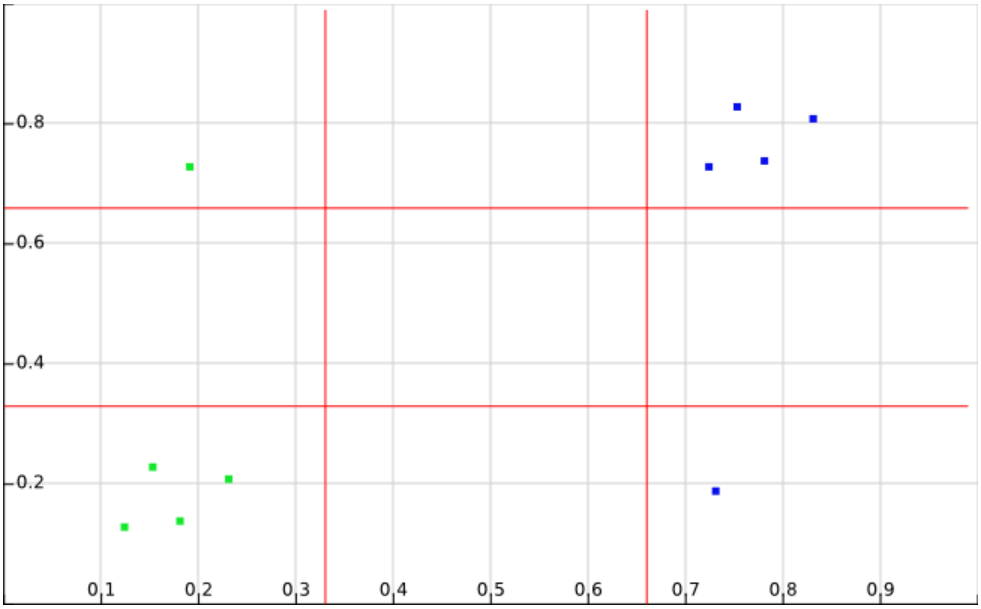


Figure 4.4: Example of a criteria customization space with clustered users

The problem that can appear is to have meaningless clusters as is shown in the figure 4.5. This will depend on how we have defined the representatives at the beginning of the k-means algorithm and how the groups defined are related with the fuzzy concept we are dealing. For instance, we could have cluster the users depending on their countries, but these clusters can be useful when you apply these algorithms over a specific fuzzy concept because it separate perfectly users with different opinions but could be completely useless for an other fuzzy concept where the users inside the same cluster have completely different opinions.

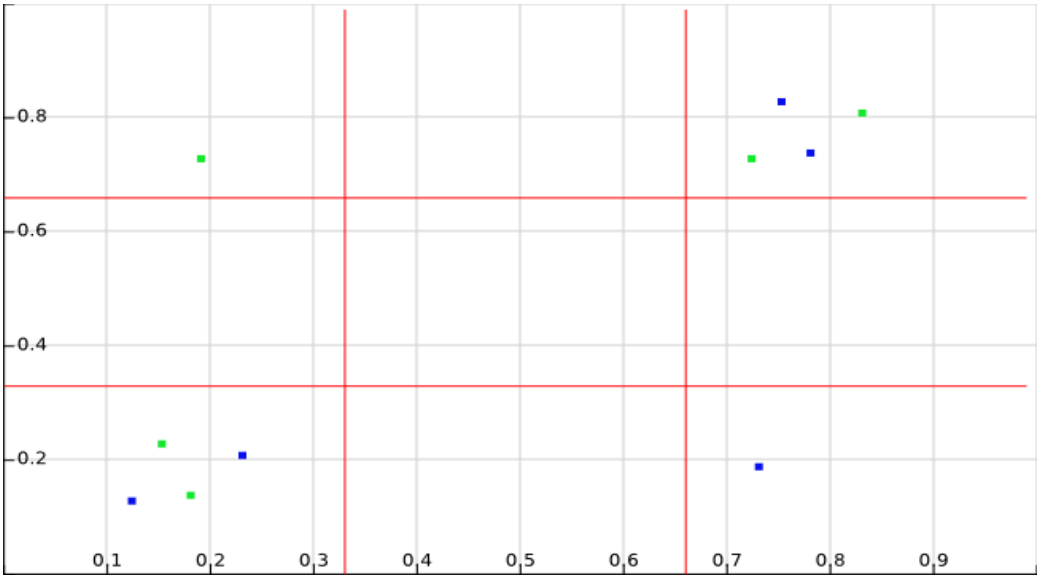


Figure 4.5: Example of a criteria customization space with clustered users

4.4. CONCLUSSIONS AND FUTURE IMPROVEMENTS

The first conclusion is that we cannot offer a unique default criteria for all the users because there will be different opinions and criteria for the same concept. Instead, we can define several groups, supposing that users from each group will have completely different opinions from the users of the rest of the groups. This idea will allow FleSe to offer a correct personalize default criteria to each user.

Then, in order to improve the precision of the clusters, the algorithm k-means will include each user in the group he belongs and, at the same time, it will automatically improve the precision of the clusters every time a new user is classified in a group, learning from the new information generated and obtained from the user.

The second algorithm will take care of defining and improving the default criteria assigned to each cluster. So every time a user makes a new customization of a criteria, the algorithm will automatically learn from it and redefined the default criteria for the group the user belongs, taking into account the new criteria customization.

As a final conclusion, it is shown that the clusters are the concept and the process that gives to FleSe the possibility to offer a personalize service to each user and allows to apply machine learning algorithms to generate useful results for each user.

As we have commented before, it would be a big improvement to make different clusters based on different characteristics depending on the Prolog file that the user wants to consult. This is because the clusters generated following a specific criteria, such as the country, could separate perfectly the users that have different opinions about a concept, but these same clusters could be completely useless for another concept.

In order to offer a more personalize service and make the information more accessible would be a great improvement to create a database to store the users information and their characteristics, and also store the information related with the algorithm k-means, such as the different the characteristics that the cluster is based on, the representatives and members of each group. With these databases you could make changes in the clusters if needed, and the information would be much more accessible.

It is also important to increase the number of characteristics that are obtained from all the login platforms used. FleSe doesn't obtain the same quantity of information from every platform and there are platforms such as Google that doesn't send any characteristic of the user.

Finally, It will be a meaningful improvement to work on the performance of the plServer. This file generate me many problems in the installation process due to the fact that its error are not managed properly and it blocks threads when a user try to load a Prolog file with errors, blocking the application when this happens multiple times.

5. BIBLIOGRAPHY

[1] V.P Ceruelo, “Extending the expressiveness of fuzzy logic lanugages”, Ph.D. disertation, Dept. Languages and Computer Science Systems and Software Engineering, UPM Univ., Madrid, ESP, 2015.

[2] Course Decision support systems, KTH, Stockholm.

[3] Course Aritifical Intelligence, UPM, Madrid.

[4] Course Declarative programming: Logic and restrictions, UPM, Madrid.

Este documento esta firmado por

	**Firmante**	CN=tfgm.fi.upm.es, OU=CCFI, O=Facultad de Informatica - UPM, C=ES
	Fecha/Hora	Mon Jan 11 03:53:53 CET 2016
	Emisor del Certificado	EMAILADDRESS=camanager@fi.upm.es, CN=CA Facultad de Informatica, O=Facultad de Informatica - UPM, C=ES
	Numero de Serie	630
	Metodo	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)